

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **2002-149350**

(43)Date of publication of application : **24.05.2002**

(51)Int.Cl. **G06F 3/06**
G06F 9/445
G06F 11/10
G06F 12/16

(21)Application number : **2000-344699** (71)Applicant : **HITACHI LTD**
HITACHI ULSI SYSTEMS
CO LTD

(22)Date of filing : **13.11.2000** (72)Inventor : **YAMATO SATOSHI**
TAKAHASHI HIROMASA
MIYAGUCHI YOSHIFUMI

(54) DISK DRIVE DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To securely perform an initialization operation decided in ATAPI specification in short time even by an application program whose program size is large.

SOLUTION: In a DVD-ROM disk drive device, a reboot program RPG, the application program APG1 and an application program APG2 are stored in a flash memory. At the time of starting the application programs APG1 and APG2, the sum value of the application program APG1 is calculated. The calculated value is collated with the previously stored sum value S1. When they are matched, the application program APG1 is started and the initialization operation is performed. Then, the sum value of the application program APG2 is calculated and the calculated value is collated with the previously stored sum value S2. When they are not matched, the application program APG2 is started.

CLAIMS

[Claim(s)]

[Claim 1] It is a disk drive device carrying a microcomputer in which it had nonvolatile memory. An application program stored in said nonvolatile memory is divided into the 1st application program and 2nd application program. Said 1st application program has the function to perform initialization provided in an ATAPI standard. It is arranged in a rewriting feasible region of said nonvolatile memory, and said 2nd application program has all the functions other than said 1st application program. It is arranged in a rewriting feasible region of said nonvolatile memory, and to a rewriting keepout area of said nonvolatile memory. A reboot program which inspects the justification of said 1st and

2nd application program is arrangedWhen said reboot program inspects the justification of said 1st and 2nd application program a sum value of a block with which said 1st application program has been arranged first is inspectedIf said 1st application program is normalafter performing initialization which made start said 1st application program and was provided in an ATAPI standard by said 1st application programA disk drive device inspecting a sum value of a block with which said 2nd application program has been arrangedand starting said 2nd application program if normal.

[Claim 2]It is a disk drive device carrying a microcomputer in which it had nonvolatile memoryAn application program stored in said nonvolatile memory is divided into the 1st application program and 2nd application programSaid 1st application program has the function to perform initialization provided in an ATAPI standardIt is arranged in a rewriting feasible region of said nonvolatile memoryand said 2nd application programIt has all the functions other than said 1st application programIt is arranged in a rewriting feasible region of said nonvolatile memoryand to a rewriting keepout area of said nonvolatile memory. A reboot program which inspects the justification of said 1st and 2nd application program is arrangedWhen said reboot program inspects the justification of said 1st and 2nd application programA sum value of a block with which said 1st application program has been arranged first is inspectedInitialization which made start said 1st application program and was provided in an ATAPI standard by said 1st application program when said 1st application program was normal is performedA disk drive device inspecting a sum value of a block with which said 2nd application program has been arrangedand starting said 2nd application program if normal if starting of said 2nd application program is needed.

[Claim 3]It is a disk drive device carrying a microcomputer in which it had nonvolatile memoryAn application program stored in said nonvolatile memory is divided into the 1st application program and 2nd application programSaid 1st application program has the function to perform initialization provided in an ATAPI standardIt is arranged in a rewriting feasible region of said nonvolatile memoryand said 2nd application programIt has all the functions other than said 1st application programIt is arranged in a rewriting feasible region of said nonvolatile memoryand to a rewriting keepout area of said nonvolatile memory. A reboot program which inspects the justification of said 1st and 2nd application program is arrangedWhen said reboot program inspects the justification of said 1st and 2nd application programA sum value of a block with which said 1st application program has been arranged first is inspectedInitialization which made start said 1st application program and was provided in an ATAPI standard by said 1st application program when said 1st application program was normal is performedIf starting of the 2nd arbitrary application program is needed among said application programsA disk drive device starting only the 2nd application program that inspected a sum value of a block with which said 2nd arbitrary application program has been arrangedand will have been arranged at said block if normal.

[Claim 4]It is a disk drive device carrying a microcomputer in which it had nonvolatile memoryIt is arranged by application program in a rewriting feasible region of said nonvolatile memoryand at the head of said application programShunting data of at least 1 byte of application program arranged at a head and an end in each block of said rewriting feasible region is arrangedA reboot program which inspects the justification of said application program to a rewriting keepout area of said nonvolatile memory is

arrangedWhen said reboot program inspects the justification of said application programIt is compared whether at least 1 byte of application program data which rewrote with said shunting data and has been arranged at a head and an end in each block of a feasible region is in agreementA disk drive device starting said application program when in agreement.

[Claim 5]It is a disk drive device carrying a microcomputer in which it had nonvolatile memoryAn application program is arranged in a rewriting feasible region of said nonvolatile memoryA reboot program which inspects the justification of said application program to a rewriting keepout area of said nonvolatile memory is arrangedWhen said reboot program inspects the justification of said application programOnly a sum value of a block with which said application program is arranged even when there is a block with which said application program is not arranged is inspectedA disk drive device starting the 2nd application program that will have been arranged at said block if normal.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]This invention about the check art of the application program in the disk drive device carrying the microcomputer containing nonvolatile memoryIt applies to the sum value check in the application program especially stored in nonvolatile memorysuch as a flash memoryand is related with effective art.

[0002]

[Description of the Prior Art]According to the place which this invention person examinedas the microcomputer in which it had the flash memoryand a disk drive device carrying what is called a microcomputer with a built-in flash memoryATAPI (Advanced Technology.) DVD-ROM (Digital Versatile.) by interface specificationssuch as Attachment Packet Interface and SCSI (Small Computer System Interface) There are a Disk-Read Only Memory drive devicea CD-ROM (Compact Disc-ROM) drive deviceetc.

[0003]For examplein the ATAPI DVD-ROM drive device carrying a microcomputer with a built-in flash memoryWhere a host computer and a drive device are connected with an ATAPI interfaceapplication program data is transmitted to a drive device from a host computerand a flash memory is rewritten. The method of performing by putting in block elimination and the writing of a flash memory for every block of a flash memory is taken.

[0004]In such a DVD-ROM drive device. It prepares for the case where accidentssuch as power supply cutoffoccur during rewriting of a flash memoryRewrite a part of block of a flash memoryconsider it as a keepout area and to this rewriting keepout area. When the justification of an application program field is inspected and the application program field is destroyedapplication program data is received from a host computerand there is a method which arranges the program which rewrites an application program field. The program arranged to this rewriting keepout area is called a reboot program.

[0005]For examplein the flash memory in which the blocks 0-3 assign the block 4 to 1 KBand it was assigned to 28 KBand the blocks 6 and 7 were assigned to 32 KBrespectivelythe reboot program is arranged at 4 KB of the blocks 0-3and this field is rewritten and it is made a keepout area.

[0006]An application program is arranged at the blocks 4-7. The sum value of the blocks 4-7 is stored in the end of the block 7. In such block assignment it is inspected whether the application program of the blocks 4-7 is the right whether a reboot program is in agreement with the sum value which calculates the sum value of the blocks 4-7 and is beforehand stored in the block 7 at the time of power-on and by comparing. When an application program is not right application program data is received from a host computer and re-writing is performed.

[0007]as art about the DVD-ROM drive device carrying such a microcomputer with a built-in flash memory etc. the art etc. which are indicated for example in the international publication WO 97/No. 38367 gazette "disk drive device and computer paraphernalia" are mentioned.

[0008]

[Problem(s) to be Solved by the Invention]However it was found out by this invention person in the above DVD-ROM drive devices that there are the following problems.

[0009]In recent years it is in the tendency for the program size of firmware to also become large with rapid-increase-izing of the demand of DVD-ROM drive devices and when the program size of this firmware becomes large the computation time of a sum value will also become long.

[0010]It is determined by the ATAPI standard that ATAPI devices such as a DVD-ROM drive device are performed to within a time [which was able to define processings (master / slave negotiation etc.) of the regulation which exists when a reset signal is received from a power up or a host computer].

[0011]However if calculation of a sum value takes time as be alike if the program size of firmware becomes large After calculating the sum value of an application program to within a time [which was provided in the ATAPI standard mentioned above] it has a problem that it will be difficult to execute the program of this application program field.

[0012]Then even if the purpose of this invention is an application program with large program size it provides the disk drive device which can perform initializing operation certainly provided in the ATAPI standard for a short time.

[0013]The other purposes and the new feature will become clear from description and the accompanying drawing of this specification along [said] this invention.

[0014]

[Means for Solving the Problem]It will be as follows if an outline of a typical thing is briefly explained among inventions indicated in this application.

[0015]Namely a disk drive device by this invention An application program stored in nonvolatile memory is divided into the 1st application program and 2nd application program This 1st application program has the function to perform initialization provided in an ATAPI standard being arranged in a rewriting feasible region of nonvolatile memory -- the 2nd application program -- this -- all the functions other than the 1st application program [have and] It is arranged in a rewriting feasible region of nonvolatile memory and to a rewriting keepout area of nonvolatile memory. A reboot program which inspects the justification of the 1st and 2nd application program is arranged When a reboot program inspects the justification of the 1st and 2nd application program A sum value of a block with which the 1st application program has been arranged first is inspected If this 1st application program is normal after performing initialization which made start the 1st application program and was provided in an ATAPI standard by the 1st

application program a sum value of a block with which the 2nd application program has been arranged is inspected. If normal, the 2nd application program will be started.

[0016] A disk drive device by this invention. An application program stored in nonvolatile memory is divided into the 1st application program and 2nd application program. This 1st application program has the function to perform initialization provided in an ATAPI standard. It is arranged in a rewriting feasible region of said nonvolatile memory, and this 2nd application program has all the functions other than the 1st application program, and it is arranged in a rewriting feasible region of nonvolatile memory. A reboot program which inspects the justification of the 1st and 2nd application program to a rewriting keepout area of this nonvolatile memory is arranged. When this reboot program inspects the justification of the 1st and 2nd application program, a sum value of a block with which the 1st application program has been arranged first is inspected. If the 1st application program is normal, initialization which made start the 1st application program and was provided in an ATAPI standard by the 1st application program is performed, and starting of the 2nd application program is needed. A sum value of a block with which the 2nd application program has been arranged is inspected, and if normal, the 2nd application program will be started.

[0017] A disk drive device by this invention. An application program stored in nonvolatile memory is divided into the 1st application program and 2nd application program. This 1st application program has the function to perform initialization provided in an ATAPI standard. It is arranged in a rewriting feasible region of nonvolatile memory, and this 2nd application program has all the functions other than the 1st application program, and it is arranged in a rewriting feasible region of nonvolatile memory. A reboot program which inspects the justification of the 1st and 2nd application program to a rewriting keepout area of nonvolatile memory is arranged. When this reboot program inspects the justification of the 1st and 2nd application program, a sum value of a block with which the 1st application program has been arranged first is inspected. Initialization which made start the 1st application program and was provided in an ATAPI standard by the 1st application program when the 1st application program was normal is performed. If starting of the 2nd arbitrary application program is needed among application programs, a sum value of a block with which the 2nd arbitrary application program has been arranged is inspected, and if normal, only the 2nd application program arranged at this block will be started.

[0018] A disk drive device by this invention. It is arranged by application program in a rewriting feasible region of nonvolatile memory, and at the head of this application program, shunting data of at least 1 byte of application program arranged at a head and an end in each block of a rewriting feasible region is arranged. A reboot program which inspects the justification of an application program to a rewriting keepout area of nonvolatile memory is arranged. When this reboot program inspects the justification of an application program, it compares whether at least 1 byte of application program data which rewrote with shunting data and has been arranged at a head and an end in each block of a feasible region is in agreement, and an application program is started when in agreement.

[0019] A disk drive device by this invention. An application program is arranged in a rewriting feasible region of nonvolatile memory. A reboot program which inspects the justification of an application program to a rewriting keepout area of this nonvolatile

memory is arrangedWhen this reboot program inspects the justification of an application programEven when there is a block with which an application program is not arrangedonly a sum value of a block with which an application program is arranged is inspectedand if normalthe 2nd application program arranged at a block will be started.

[0020]

[Embodiment of the Invention]Hereafteran embodiment of the invention is described in detail based on a drawing.

[0021](Embodiment 1) The functional block diagram of the disk drive device by the embodiment of the invention 1 and drawing 2 drawing 1The explanatory view and drawing 3 in which the memory map of the flash memory with which the microcomputer provided in the disk drive device by the embodiment of the invention 1 was equipped is shown are a flow chart which shows the activation procedure of the program in the flash memory at the time of the startup by the embodiment of the invention 1.

[0022]In this Embodiment 1a disk drive device is an ATAPI DVD-ROM drive device by which a microcomputer with a built-in flash memory is carriedfor example.

[0023]This disk drive device comprises the DVD-ROM playback equipment 2 etc. which play the information memorized by the DVD-ROM disk 1 with which information is memorizedand this DVD-ROM disk 1.

[0024]Via the ATAPI busthe DVD-ROM playback equipment 2 is bidirectionally connected at the host computer 3 so that a signal transmission is possible. The DVD-ROM playback equipment 2 comprises microcomputer 4digital signal processing part 5preamplifier 6disk motor 7Motor Driver 8pickup 9and cache DRAM10 etc.

[0025]The DVD-ROM disk 1 is rotated with the disk motor 7. The DVD-ROM disk 1 to rotate is irradiated with a laser beamthe catoptric light is received by the light sensing portion which consists of photo-diodesoptical conversion is carried outand the pickup 9 reads the information memorized by the DVD-ROM disk 1. Motor Driver 8 controls the revolving speed of the disk motor 7.

[0026]The digital signal processing part 5 comprises the data strobe circuit 11the servo circuit 12the digital disposal circuit 13 for DVD RAM controller 14the digital disposal circuit 15 for CDthe ATAPI interface circuitry 16etc.

[0027]The data strobe circuit 11 comprises a data slicing circuita PLL (Phase Locked Loop) circuitetc.A binary (digital) turns the read signal (high frequency signal) which was read from the pickup 9 and amplified by the preamplifier 6and a PLL clock is outputted based on the binary-ized data.

[0028]The servo circuit 12 controls the thread motor with which it prepared for the pickup 9and controls the position of this pickup 9. RAM controller 14 performs access control to cash DRAM10.

[0029]The ATAPI interface circuitry 16 to the host computers 3such as a personal computer. It is the interface used when connecting apparatussuch as a disk drive device to an ATA interfaceand interface control based on an ATAPI interface is performed.

[0030]The digital disposal circuit 13 for DVD and the digital disposal circuit 15 for CD have an EFM demodulation functiona sub-code demodulation functionan error correction functionetc.respectively. It restores to an EFM demodulation function in the EFM (Eight-to-Fourteen Modulation) abnormal conditions read signal based on the synchronized signal detected by the speed-control function.

[0031]A sub-code demodulation function restores to the sub-code contained in each

frame of the read signal to which it restored to the EFM demodulation function. An error correction function is an error correction performed using an error correcting code called CIRC which combined the Reed Solomon code of two series of C1 and C2 and the error correcting code corresponds to the parity of a frame.

[0032]The recorded information to which it was binary-ized by the data strobe circuit 11 and the digital signal was processed by the EFM demodulation function and the sub-code demodulation function and C1 and C2 correction were carried out is held by control of RAM controller 14 cash DRAM10.

[0033]The read data held this cache DRAM10 is outputted to the host computer 3 from the ATAPI interface circuitry 16 per the singular number or two or more sectors.

[0034]The microcomputer 4 comprises circuit modules such as the flash memory (nonvolatile memory) 17, RAM 18, a processor, 8 bit timers, 16 bit timers, an A/D converter, an SCI circuit, and input/output port. These circuit modules share the internal bus which consists of an address bus, a data bus, and a control signal bus.

[0035]The flash memory 17 is nonvolatile memory in which electric elimination and writing are possible and is a memory for storing the operation program and constant data of a processor. Collective erasure of the storage area is made possible by a predetermined block unit and the flash memory 17 has the composition which can rewrite data by a block unit.

[0036]A processor controls the preamplifier 6, the digital signal processing part 5, etc. according to the operation program. RAM 18 is set to work RAM used for the work region of a processor, etc.

[0037]The access control signal for a processor to access the digital signal processing part 5 is constituted so that a direct output may be carried out from this processor. The processor can set control information as RAM controller 14 and can access direct cash DRAM10 via RAM controller 14.

[0038]In the DVD-ROM drive device constituted in this way, reboot program RPG and application program APG1 divided into two and APG2 are stored in the flash memory 17.

[0039]Next, the method of assigning in the flash memory 17 of this embodiment is explained using the explanatory view of the memory map of drawing 2.

[0040]As the memory space of the flash memory 17 is 256 KB and it is here shown in drawing 2, block EB0 rewrites and 32 KB and the blocks EB9-EB11 are assigned [the blocks EB0-EB7] for 4 KB and block EB8 to 64 KB respectively and it is made [it is a keepout area, the blocks EB2-EB11 rewrite and] the feasible region.

[0041]Reboot program RPG is stored in block EB0. Application program (1st application program) APG1 is stored in block EB1 and EB2 and application program (2nd application program) APG2 is stored in the blocks EB3-EB11.

[0042]Reboot program RPG is the justification of application program APG1 and APG2, a program to inspect and application program APG1. For example, it is an initial operation program which must be executed within a certain prescribed period after reset by powering on such as processing of a negotiation, etc. Application program APG2 are all the programs other than application program APG1 mentioned above.

[0043]At the last (block EB2) of application program APG1. The sum value S1 in this application program APG1 is stored and the sum value S2 in this application program APG2 is stored in the last (block EB11) of application program APG2.

[0044]Next, the activation procedure of the program in the flash memory 17 at the time of

starting is explained using drawing 2 and the flow chart of drawing 3.

[0045]First if reset by the reset signal or the reset signal from the host computer 3 is canceled by powering on to a disk drive device (Step S101) The sum value of block EB1 and application program APG1 which were stored in EB2 is computed (Step S102).

[0046]And it is judged whether the computed value and the sum value S1 stored in block EB2 are compared (Step S103) and the computed sum value and the sum value S1 stored are in agreement (Step S104).

[0047]In processing of this step S104 when a computed value and the sum value S1 are in agreement application program AGP1 which is an initial operation program is started and initialization provided in the ATAPI standard is performed (Step S105).

[0048]When a computed value and the sum value S1 are not in agreement with Step S104 in processing rewriting of application program APG1 is performed to block EB1 and EB2 (Step S106).

[0049]After processing of Step S105 is completed the sum value of application program APG2 stored in the blocks EB3-EB11 is computed (Step S107) and the computed value and the sum value S2 stored in block EB2 are compared (Step S108).

[0050]And when it judges whether the computed sum value and the sum value S2 stored are in agreement (Step S109) and the sum value S2 and a computed value are in agreement application program AGP2 starts (Step S110) and it functions as a disk drive device.

[0051]Here application program APG2 Without performing a sum value check promptly after execution of application program APG1 after application program APG2 is needed it may be made to perform the sum value check of this application program APG2 for example.

[0052]Or the sum value check of only the block with which application program APG2 which is needed among application program APG2 was stored may be performed.

[0053]In processing of Step S109 when the sum value and the sum value S2 which were computed are not in agreement rewriting of application program APG2 is carried out to the blocks EB3-EB11 (Step S106).

[0054]According to this Embodiment 1 an application program is divided into two by that cause The sum value of application program AGP1 which performs initialization provided in the ATAPI standard which needs to be processed at the short time after reset is given priority to and checked By performing only application program AGP1 first Even if the program size of an application program becomes large initialization processing specified to the ATAPI standard can be certainly performed within a prescribed period and the reliability of a disk drive device can be raised.

[0055]Application programs run APG1 which performs initialization in which the application program was provided by the ATAPI standard after reset in this Embodiment 1 When it is divided into application program AGP2 which are all the programs other than this application program APG1 and a sum value was checked attached and indicated but. When there is a free space where the application program is not stored in the flash memory the time of a sum value check can be shortened also by being made not to perform the sum value check in the block of these free space.

[0056]As the memory space of a flash memory is 256 KB and it is shown in drawing 4 32 KB and the blocks EB9-EB11 are assigned [the blocks EB0-EB7] for 4 KB and block EB8 to 64 KB respectively block EB0 rewrites it shall be a keepout area the blocks EB2-

EB11 shall rewrite and it shall be made the feasible region.

[0057] And reboot program RPG is stored in block EB0. When application program APG3 is stored in the blocks EB1-EB10 and the block EB11 [remaining] has become a free space after release of reset. Only the sum value of application program APG3 stored in the blocks EB1-EB10 is checked and it can set to block EB11 and is made not to perform R8 sum-value check.

[0058] For example, since the memory space of a flash memory runs short in 128KB when the program size of an application program is about 150 KB, the flash memory whose memory space is 256 KB will be used.

[0059] In such a case, since the block of about 100 KB serves as a free space and a sum value check can be made unnecessary, the time of the sum value check of application program APG3 can be shortened substantially.

[0060] Also, by this initialization processing specified to the ATAPI standard can be certainly performed within a prescribed period and the reliability of a disk drive device can be raised.

[0061] (Embodiment 2) Drawing 5 is an explanatory view showing the memory map of the flash memory with which the microcomputer provided in the disk drive device by the embodiment of the invention 2 was equipped.

[0062] In this Embodiment 2, a disk drive device (drawing 1) is an ATAPI DVD-ROM drive device by which a microcomputer with a built-in flash memory is carried like said Embodiment 1. It comprises the DVD-ROM disk 1, the DVD-ROM playback equipment 2 bidirectionally connected via the ATAPI bus at the host computer 3 so that a signal transmission was possible, etc.

[0063] The DVD-ROM playback equipment 2 as well as said Embodiment 1 comprises microcomputer 4, digital signal processing part 5, preamplifier 6, disk motor 7, Motor Driver 8, pickup 9, and cache DRAM 10, etc.

[0064] The internal configuration of the microcomputer 4 as well as Embodiment 1 comprises circuit modules such as the flash memory 17, RAM 18, a processor 8, bit timers 16, bit timers, an A/D converter, an SCI circuit, and input/output port. These circuit modules are sharing the internal bus which consists of an address bus, a data bus, and a control signal bus.

[0065] Reboot program RPG and application program APG4 are stored in the flash memory 17. As the memory space of the flash memory 17 is 256 KB and it is shown in drawing 5, Block EB0 rewrites 32 KB and the blocks EB9-EB11 are assigned [the blocks EB0-EB7] for 4 KB and block EB8 to 64 KB, respectively, and it is made [it is a keepout area, the blocks EB2-EB11 rewrite and] the feasible region.

[0066] Reboot program RPG is stored in block EB0. Application program APG4 is stored in the blocks EB1-EB11.

[0067] Reboot program RPG is the justification of application program APG4, a program to inspect and application program APG4. They are all the application programs in a disk drive device which contain the initial operation program which must be executed within a certain prescribed period after reset by powering on, such as processing of the application program negotiation of a disk drive device, etc.

[0068] To head [of application program APG4] i.e. block EB1. Copy data (shunting data) CD about several bytes (at least 1 byte) of the head of application program APG4 and an end stored in each block EB1-EB11 is stored.

[0069]And if reset by the reset signal after the power supply was supplied to the disk drive device or the reset signal from the host computer 3 is canceled Every blocks EB1-EB11 several bytes of program data of a head and an end set up beforehand is read and it compares with copy data CD stored in the head of application program APG4.

[0070]When several of these bytes of program data and copy data CD are altogether in agreement in each block EB1-EB11 application program AGP3 is started and performed.

[0071]When there is a block several bytes of program data and whose copy data CD do not correspond application program APG4 is rewritten by a block unit.

[0072]By that cause also in this Embodiment 2 since the sum check of application program APG4 becomes unnecessary The check of application program APG4 can be ended in a short time even if the program size of application program APG4 becomes large initialization provided in the ATAPI standard which needs to be processed at the short time after reset is performed -- it can perform within a prescribed period certainly and the reliability of a disk drive device can be raised.

[0073]As mentioned above although the invention made by this invention person was concretely explained based on the embodiment it cannot be overemphasized that it can change variously in the range which this invention is not limited to said embodiment and does not deviate from the gist.

[0074]For example in said embodiment although the ATAPI DVD-ROM drive device and the SCSI DVD-ROM drive device were explained to the example It is effective for not the thing limited to this but an ATAPI CD-ROM drive device an ATAPI CD-RW drive device a SCSI DVD-ROM drive device a SCSI CD-ROM drive device a SCSI CD-RW drive device etc.

[0075]

[Effect of the Invention] It will be as follows if the effect acquired by the typical thing among the inventions indicated by this application is explained briefly.

[0076](1) By giving priority to and performing only the 1st application program that performs initialization provided in the ATAPI standard which needs to be processed at the short time after reset according to this invention Even if program size becomes large initialization processing specified to the ATAPI standard can be certainly performed within a prescribed period.

[0077](2) In this invention the time of the sum value check of this application program can be substantially shortened by starting an application program without performing the sum value check of the block in a free space.

[0078](3) Shunting data of at least 1 byte of application program further arranged in this invention at the head and end in each block of a rewriting feasible region Since at least 1 byte of application program data arranged at the head and end in each block is compared and this application program is started when in agreement The sum check of an application program becomes unnecessary and the check of an application program can be ended in a short time.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is a functional block diagram of the disk drive device by the embodiment

of the invention 1.

[Drawing 2] It is an explanatory view showing the memory map of the flash memory with which the microcomputer provided in the disk drive device by the embodiment of the invention 1 was equipped.

[Drawing 3] It is a flow chart which shows the activation procedure of the program in the flash memory at the time of starting by the embodiment of the invention 1.

[Drawing 4] It is an explanatory view showing an example of the memory map of the flash memory with which the microcomputer provided in the disk drive device by other embodiments of this invention was equipped.

[Drawing 5] It is an explanatory view showing the memory map of the flash memory with which the microcomputer provided in the disk drive device by the embodiment of the invention 2 was equipped.

[Description of Notations]

1 DVD-ROM disk

2 DVD-ROM playback equipment

3 Host computer

4 Microcomputer

5 Digital signal processing part

6 Preamplifier

7 Disk motor

8 Motor Driver

9 Pickup

10 Cash DRAM

11 Data strobe circuit

12 Servo circuit

13 The digital disposal circuit for DVD

14 RAM controller

15 The digital disposal circuit for CD

16 ATAPI interface circuitry

17 Flash memory (nonvolatile memory)

18 RAM

APG1 Application program (the 1st application program)

APG2 Application program (the 2nd application program)

APG3 Application program

APG4 Application program

RPG Reboot program